■ ประพนธ์ อัศวกาญวัฒน์

สร้างเกมส์ต่อภาพ Jigsaw Hunter ด้วยตัวเอง ตอนที่ 1

เล่นเกมส์มาก็มากมายแล้ว
ลองนำไอเดียจากเกมส์ที่เล่น
มาสร้างเกมส์เอง
ไม่ยากอย่างที่คิดไว้

Delphi

สวัสดีครับ ปัจจุบันธุรกิจซอฟต์แวร์ด้านความบันเทิงถือเป็นตลาดแห่งใหม่ที่น่าสนใจ จำนวนเม็ดเงินมหาศาลและผลกำไรในธุรกิจช่วยผลักดันให้ธุรกิจนี้เติบโตรวดเร็วเป็นอันมาก แม้กระทั่งบริษัทที่เป็นเจ้าแห่งซอฟต์แวร์ยักษ์ใหญ่ทางด้านธุรกิจอย่างไมโครซอฟท์ยังอดไม่ได้ที่ต้องเข้าร่วมวงไพบูลย์ ด้วยการทำหน้าที่เป็นทั้งผู้พัฒนาและผู้จัดจำหน่ายให้กับเกมส์ดัง ๆ มากมาย ทั้งนี้ยังไม่รวมถึงเครื่อง "เอ็กซ์บ็อกซ์" (X-BOX) ที่ไมโครซอฟท์ปลุกปั้นด้วยเทคโนโลยีของเครื่องไมโครคอมพิวเตอร์ล่าสุดเพื่อต่อกรกับเครื่องเล่นเกมส์คอนโซล (Game Console) ทั้งหลายในตลาดดั้งเดิม

ในฉบับนี้ผู้เขียนจะขอกล่าวถึงการพัฒนาโปรแกรมอีกประเภทหนึ่งที่มีจุดประสงค์เพื่อความบันเทิงแทนการใช้งาน คือการพัฒนาโปรแกรมชนิดเกมส์ (Game Programming) เพื่อการเรียนรู้หลักการและแนวความคิดในการพัฒนาโปรแกรมอีกรูปแบบที่แตกต่างจากโปรแกรมใช้งานแบบเดิม สำหรับโปรแกรมที่นำมาศึกษานี้จะใช้คำสั่งมาตรฐานของระบบวินโดวส์เป็นเกณฑ์ เพื่อให้สามารถเรียนรู้และสร้างโปรแกรมได้อย่างรวดเร็ว

โปรแกรมเกมส์และเทคโนโลยี

ก่อนจะเริ่มต้นพัฒนาโปรแกรมชนิดนี้ ผู้เขียนจะให้ผู้อ่านทุกท่านทำความรู้จักกับจุดมุ่งหมายของโปรแกรมชนิดนี้ก่อน กล่าวคือ โปรแกรมเกมส์หรือเกมส์คอมพิวเตอร์เป็นรูปแบบของโปรแกรมชนิดหนึ่งที่มีจุดประสงค์ให้ผู้ใช้หรือผู้เล่นตอบสนองต่อเหตุการณ์ต่าง ๆ ที่โปรแกรมสร้างขึ้น เพื่อให้โปรแกรมสามารถดำเนินต่อไปจนบรรลุจุดประสงค์ของเกมส์ที่กำหนดไว้ และดึงดูดความสนใจของผู้เล่น โปรแกรมเกมส์จึงเป็นโปรแกรมที่ได้รวบรวมเทคโนโลยีต่าง ๆ รวมทั้งระบบมัลติมีเดีย (Multimedia) ที่ต้องประกอบด้วยระบบเสียง (Sound & Music) ภาพเคลื่อนไหว (Animation) เพื่อสร้างความสนใจชวนให้ติดตามและตอบสนองต่อเกมส์นั้น ๆ

อาจกล่าวได้ว่าเทคโนโลยีที่ใช้ในการพัฒนาโปรแกรมเกมส์เป็นแบบก้าวกระโดด สามารถสังเกตจากเกมส์บนระบบคอมพิวเตอร์ในอดีตที่สร้างขึ้นเพื่อความเพลิดเพลินบนคอมพิวเตอร์แพลตฟอร์ม (Platform) ใดแพลตฟอร์มหนึ่ง ซึ่งมีเพียงภาพแสดงผลแบบโมโนโครม (Monochrome) และการควบคุมรูปภาพหรือสไปซ์ (Sprite) ให้ตอบสนองต่อเหตุการณ์ที่เกิดขึ้นอย่างง่าย ๆ ไม่ซับซ้อน เช่น เกมส์ PONG ซึ่งเป็นการบังคับรูปภาพแท่งไม้ (Stick) คอยไล่ตีรูปภาพลูกบอล (Ball) เพื่อให้กระดอนไปมา

แต่ในปัจจุบันการพัฒนาเกมส์แตกต่างออกไปจนกลายเป็นอุตสาหกรรมขนาดใหญ่ มีการวางแผนและเตรียมการเพื่อสนองความต้องการกับลูกค้าทุกกลุ่มเป้าหมายในแทบทุกระบบคอมพิวเตอร์ที่เอื้ออำนวย โดยการทุ่มทุนคิดค้นเทคโนโลยีต่าง ๆ เพื่อช่วยเสริมให้เกมส์มีความซับซ้อน และการสนองตอบที่ใกล้เคียงความเป็นจริงมากขึ้น เช่น ระบบแสดงผลแบบ 3D ช่วยให้สามารถแสดงผลได้รวดเร็ว สามารถให้แสงเงาและพื้นผิวสมจริงมากยิ่งขึ้น รวมทั้งการสร้างสถานการณ์โดยใช้กฎฟิสิกส์จำลองสภาวะเสมือน ให้ผู้เล่นมีความรู้สึกได้ร่วมอยู่ในเหตุการณ์จริง

ชนิดของเกมส์

จากแนวความคิดในการตอบสนองความต้องการของผู้เล่นทุกกลุ่มเป้าหมาย ทำให้อุตสาหกรรมเกมส์ที่พัฒนาขึ้นถูกแบ่งแยกเป็นหมวดหมู่ใหญ่ ๆ เพื่อตอบสนองกลุ่มผู้เล่น ซึ่งพอจะสรุปได้ดังนี้คือ

Action Game เป็นเกมส์ที่อาศัยความสัมพันธ์ของประสาทสัมผัสของสายตาและมือ เพื่อตอบสนองต่อเหตุการณ์ที่เกิดขึ้น เกมส์แนวนี้เน้นการต่อสู้และการตอบสนองอย่างรวดเร็วเป็นหลัก เพื่อให้สามารถดำเนินเกมส์ตามเนื้อเรื่องให้ได้มากที่สุดหรือจนจบเกมส์ ตัวอย่างของเกมส์ประเภทนี้ในปัจจุบันคือ Half-Life, Quake, Sirous Sam และ Unreal เป็นต้น

Adventure Game เป็นเกมส์ที่กำหนดให้ผู้เล่นต้องใช้ความคิดในการเอาตัวรอด โดยกำหนดให้ผู้เล่นสวมบทบาทเข้าสู่สถานการณ์ที่กำหนด เพื่อแก้ไขปริศนา สำรวจตามเรื่องราว และเนื้อหาที่กำหนด เกมส์ประเภทนี้ส่วนใหญ่จึงเป็นเกมส์ที่ใช้เนื้อเรื่องเป็นตัวกำหนดเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ทำให้เกมส์ประกอบด้วยทักษะต่าง ๆ ของเกมส์ชนิดอื่น ๆ ผสมผสานร่วมอยู่ด้วย ตัวอย่างเกมส์ชนิดนี้คือ Alone in the dark, Resident Evil และ Tomb Raider เป็นต้น

Puzzle Game หรือลอจิกเกมส์เป็นเกมส์ที่ผู้เล่นต้องใช้ความคิดในการแก้ไขลอจิกหรือปัญหาที่เกมส์กำหนดขึ้น ลักษณะของเกมส์จะมุ่งเน้นให้ผู้เล่นใช้ความคิดในการแก้ไขปัญหาของเกมส์เป็นหลัก โดยไม่จำเป็นต้องมีเนื้อเรื่องกำหนดการดำเนินเรื่องราวของเกมส์ ความสนุกของเกมส์ชนิดนี้จะเกิดจากความยากของปัญหาที่เพิ่มขึ้นเรื่อย ๆ ตัวอย่างของเกมส์ชนิดนี้คือ Card Game, Jigsaw, Minesweeper และ Tetris เป็นต้น

Role-Playing Game หรือเรียกย่อว่าเกมส์ RPG เป็นรูปแบบของเกมส์ที่ดำเนินตามเนื้อเรื่อง มีตัวละครในเกมส์จำนวนมากสร้างความสัมพันธ์กันทั้งที่ผู้เล่นควบคุมเองและคอมพิวเตอร์ควบคุม ลักษณะพิเศษของเกมส์ชนิดนี้คือตัวละครสามารถพัฒนาทักษะต่าง ๆ ได้ ส่งผลให้ผลลัพธ์ของการดำเนินเกมส์ตามเนื้อเรื่องแตกต่างกันออกไป ตัวอย่างเกมส์ชนิดนี้ เช่น Final Fantasy และ Ultima เป็นต้น

Simulations Game เป็นเกมส์แนวจำลองสถานการณ์ต่าง ๆ ให้มีความสมจริงมากที่สุด โดยเน้นการกำหนดขั้นตอนในการปฏิบัติตามหลักความเป็นจริงด้วยสภาวะจำลอง เช่น เกมส์จำลองการขับเครื่องบิน เกมส์ชนิดนี้จึงเป็นต้นแบบของโปรแกรมฝึกหัดทักษะให้กับผู้ปฏิบัติโดยไม่ต้องปฏิบัติจริง ตัวอย่างเกมส์ชนิดนี้คือ Fleet Command, Jack Nicklaus, The Sims และ World War II Fighter เป็นต้น

Sport Game เป็นเกมส์ที่ดำเนินเรื่องด้วยกีฬาเป็นหลัก ซึ่งอาจจะเป็นเกมส์จำลองการแข่งขันกีฬาประเภทต่าง ๆ เช่น การแข่งขันฟุตบอล การแข่งขันกอล์ฟ เป็นต้น ตัวอย่างของเกมส์ชนิดนี้คือ FIFA 2000 และ NBA LIVE 2000

Strategy Game เป็นเกมส์ที่ต้องใช้ความคิดในการวางแผนในการควบคุมทรัพยากร และเวลาที่มีอยู่อย่างจำกัดให้ได้ประโยชน์สูงสุด เกมส์ชนิดนี้จึงเน้นการใช้ทักษะในการคิดและวางแผน เพื่อให้สามารถชนะคู่ต่อสู้หรือฝ่ายตรงกันข้ามให้ได้รวดเร็วที่สุด ตัวอย่างเกมส์ประเภทนี้ในปัจจุบันคือ C&C : Red Alert, Pharaoh และ Star Craft เป็นต้น

การพัฒนาเกมส์

ในการเริ่มต้นพัฒนาเกมส์ผู้เขียนขอเลือกพัฒนาโปรแกรมเกมส์ประเภท Puzzle Game เนื่องจากการพัฒนาเกมส์หนึ่ง ๆ นั้นสิ่งสำคัญอย่างมากที่จะทำให้เกมส์ประสบความสำเร็จคือเทคโนโลยีและรูปแบบเนื้อหาของเกมส์ การที่ผู้เขียนเลือกพัฒนาโปรแกรมเกมส์ประเภท Puzzle Game ทำให้ผู้เขียนสามารถใช้เวลาทั้งหมดในการสร้างสรรค์เนื้อหาของเกมส์ โดยไม่ต้องเสียเวลาเรียนรู้การใช้งานเทคโนโลยีต่าง ๆ ซึ่งจุดนี้ผู้เขียนคิดว่าผู้ที่อยู่ในอุตสาหกรรมเกมส์จริง ๆ เหมาะสมมากกว่า

เมื่อพิจารณาโปรแกรมเกมส์ชนิด Puzzle Game ด้วยเทคโนโลยีพื้นฐานของระบบวินโดวส์ สามารถสร้างโปรแกรมเกมส์อะไรได้บ้าง สิ่งที่ทำได้คงเป็นโปรแกรมเกมส์ที่มีการเคลื่อนไหวไม่ซับซ้อนเกินไป ผู้เขียนจึงเลือกที่จะพัฒนาโปรแกรมเกมส์ต่อภาพ (Jigsaw Puzzle) เหตุผลเสริมที่ทำให้ผู้เขียนเลือกพัฒนาโปรแกรมเกมส์ประเภทนี้คือเรื่องของทรัพยากร (Resource) ไม่ว่าจะเป็นรูปภาพที่ใช้หรือเสียงประกอบ เพราะผู้เขียนสามารถค้นหาสิ่งเหล่านี้ได้โดยตรงจากระบบอินเทอร์เน็ต รวมทั้งไม่ต้องเสียเวลาในการจัดเตรียมมากนัก

สิ่งที่ทุกท่านควรทราบก่อนเริ่มต้นพัฒนาเกมส์ใด ๆ ที่สนใจ ซึ่งผู้เขียนพบในการพัฒนาเกมส์นี้ให้สำเร็จมีดังนี้

- การเริ่มต้นในสิ่งที่ท้าทายเป็นแรงบันดาลใจที่ช่วยชักจูงให้เกิดความคิดในการสร้างสรรค์สิ่งต่าง ๆ ได้อย่างง่ายดาย และอุปสรรคจะเกิดขึ้นทันทีที่เริ่มต้นสร้างสรรค์สิ่งนั้น ที่ผู้เขียนกล่าวเช่นนี้เนื่องจากความรู้สึกของตนเองในตอนเริ่มต้นสร้างเกมส์นั้นค่อนข้างจะสนุกและอยากจะทำให้เกมส์ออกมาสวยงามน่าสนใจ แต่เมื่อลงมือกระทำแล้วจะพบคำถามมากมาย เช่น ทำอย่างไรให้เกมส์สวยงาม เป็นต้น

- ความสามารถเป็นสิ่งที่เรียนรู้กันได้ แต่พรสวรรค์มิได้มีอยู่ในทุกคน ซึ่งผู้เขียนคิดว่าสิ่งนี้เป็นสิ่งที่แบ่งแยกระดับขั้นของคำว่า "เก่งกาจ" กับ "อัจฉริยะ" ออกจากกันอย่างสิ้นเชิง ในโลกนี้ไม่ง่ายนักที่จะหาคนอย่าง John Carmack หรือ Sid Meier ในการสร้างสรรค์ผลงานเกมส์ที่ดีได้ ดังนั้นไม่ว่าเกมส์ของท่านจะออกมาเช่นไร อย่าเพิ่งท้อแท้และหยุดพัฒนากลางคัน เพราะสิ่งที่ท่านกำลังตั้งใจทำคือการเรียนรู้

แนวความคิดในการสร้างเกมส์ต่อภาพ

เกมส์ต่อภาพหรือ Jigsaw Puzzle เป็นเกมส์การจัดเรียงส่วนของรูปภาพที่คละกันให้เป็นรูปภาพสมบูรณ์ ในการพัฒนาโปรแกรมเกมส์ตามความหมายข้างต้นเพียงอย่างเดียว เกมส์นี้อาจจะไม่มีผู้สนใจเลย เนื่องจากไม่มีจุดดึงดูดให้น่าสนใจ ดังนั้นการสร้างเกมส์ให้มีความน่าสนใจจึงเป็นจุดมุ่งหมายหลักที่ต้องทำมากกว่าการเขียนโปรแกรมเกมส์ให้ดีเยี่ยมเสียอีก

ในส่วนนี้ผู้เขียนได้กำหนดความน่าสนใจของเกมส์ด้วยการเพิ่มชีวิตชีวาให้กับเกมส์ โดยแต่งเติมส่วนประกอบต่าง ๆ ที่เกิดขึ้นขณะเล่นเกมส์ดังนี้

- เพลงประกอบ คือเสียงเพลงที่จะบรรเลงในขณะที่เกมส์กำลังดำเนินอยู่ ซึ่งแบ่งออกเป็นสองระดับ กล่าวคือ ระดับปกติ (Normal Music) ใช้บรรเลงในช่วงเวลาที่เกมส์กำลังดำเนินหรือโปรแกรมเกมส์กำลังทำงาน และระดับเร่งเร้า (Hurry up) ใช้กระตุ้นให้ผู้เล่นเกมส์รู้สึกเร่งรีบ
- เสียงประกอบ เป็นเสียงที่บันทึกเพื่อให้เกิดการตอบสนองการดำเนินการของเกมส์ในแต่ระดับขั้น รวมทั้งใช้เป็นสื่อสำหรับสื่อความหมายหรือแนะนำการกระทำของผู้เล่นที่มีต่อเกมส์
- เครื่องมือวิเศษ เป็นส่วนของโปรแกรมที่กำหนดความสามารถเสมือนอุปกรณ์วิเศษที่ช่วยให้ผู้เล่นสามารถแก้ไขปัญหาได้รวดเร็วหรือง่ายยิ่งขึ้น โดยการใช้งานเครื่องมือวิเศษทั้งหมดต้องอยู่ภายใต้เงื่อนไขของเวลาและระดับขั้นความยากของเกมส์ ซึ่งประกอบด้วย

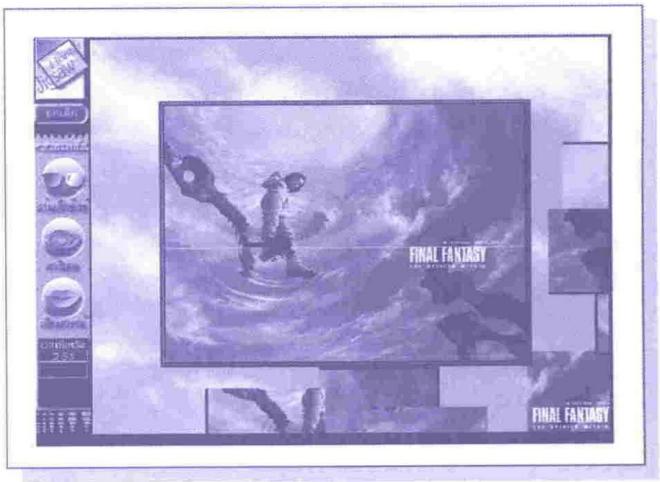
1. ตาเอ็กซเรย์ จะช่วยให้ผู้เล่นทราบถึงตำแหน่งที่ถูกต้องแน่นอนของชิ้นภาพโดยมองเห็นตำแหน่งและลำดับที่ถูกต้องของชิ้นภาพ
2. ตาวิเศษ ช่วยให้ผู้เล่นเกมส์สามารถมองเห็นรูปภาพที่สมบูรณ์ของชิ้นภาพ
3. เสียงสวรรค์ เป็นเสียงกระซิบที่คอยบอกใบ้ในการวางชิ้นภาพของผู้เล่นว่าถูกต้องหรือไม่

- คะแนนรวม เป็นค่าคะแนนรวมที่ได้รับจากความรวดเร็วในการเล่นเกมส์ โดยคำนวณจากความถูกต้องของการวางชิ้นภาพและเวลาที่เหลือจากการต่อภาพ

จุดประสงค์ของการเล่นเกมส์นี้จึงเป็นการจัดเรียงส่วนต่าง ๆ ของรูปภาพที่คละเคล้ากันให้เป็นรูปภาพสมบูรณ์โดยใช้เวลาน้อยที่สุดและสามารถใช้เครื่องมือวิเศษต่าง ๆ ที่มีอยู่อย่างจำกัดช่วยแก้ไขปัญหาเมื่อไม่สามารถต่อภาพให้สมบูรณ์ได้

เพื่อให้เกิดความหลากหลายในการเล่นเกมส์ ผู้เขียนขอกำหนดเงื่อนไขของโปรแกรมเกมส์เพิ่มเติมดังนี้

1. สามารถกำหนดความยากง่ายในการเล่นเกมส์ได้ 4 ระดับคือ Basic Game, Kid Game, Normal Game และ Expert Game โดยรูปแบบของ Basic Game จะเป็นเกมส์ที่มีเงื่อนไขและข้อจำกัดน้อยที่สุด
2. สามารถกำหนดและเปลี่ยนแปลงเงื่อนไขหรือข้อจำกัดต่าง ๆ ให้เหมาะสมกับทักษะของผู้เล่นเกมส์ได้
3. สามารถแสดงบอร์ดคะแนนรวมโดยเรียงจากผู้ที่ทำคะแนนได้สูงสุด

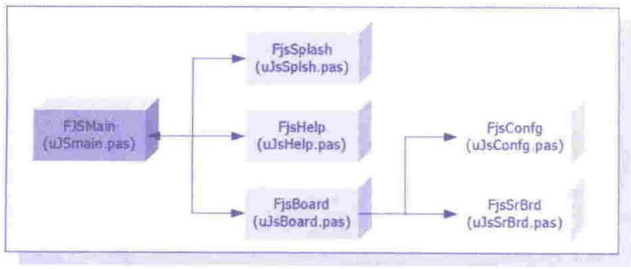


รูปที่ 1 โปรแกรมเกมส์ Jigsaw Hunter

โปรแกรมเกมส์ต่อภาพ Jigsaw Hunter

เกมส์ต่อภาพ Jigsaw Hunter ดังรูปที่ 1 เป็นเกมส์ชนิด Jigsaw Puzzle ที่สร้างขึ้นตามแนวความคิดในการสร้างเกมส์ต่อภาพข้างต้น โดยแบ่งส่วนการทำงานหลักของโปรแกรมออกเป็น 6 ส่วนตามหน้าที่ของโปรแกรมในแต่ฟอร์ม ซึ่งมีความสัมพันธ์กันดังแสดงไว้ในรูปที่ 2 ฟอร์มทั้งหมดประกอบด้วย

1. ส่วนของโปรแกรมหลัก หน้าต่าง FJSMain (uJSMain.pas) เป็นฟอร์มหลักใช้สำหรับควบคุมการทำงานของเกมส์ โดยใช้เป็นฟอร์มที่รวบรวมแบบข้อมูล ค่าคงที่ในโปรแกรมย่อยต่าง ๆ รวมทั้งส่วนของโปรแกรมควบคุมเสียงเพลงและเสียงประกอบภายในเกมส์
2. ส่วนปรับแต่งข้อกำหนด หน้าต่าง FJSConfg (uJSConfg.pas) เป็นฟอร์มที่ใช้กำหนดและปรับแต่งค่าความเหมาะสม ความยากง่ายของเกมส์ระดับต่าง ๆ
3. หน้าต่างแสดงคำแนะนำในการเล่น FjsHelp (uJSHelp.pas) เป็นฟอร์มที่ใช้แสดงคำแนะนำในการเล่นเกมส์
4. หน้าต่างแสดง High Score FfsSrBrd (uJsSrBrd.pas) เป็นฟอร์มที่ใช้แสดงรายการคะแนนของผู้เล่นที่บันทึกไว้ 12 ลำดับ
5. หน้าต่าง Splash Form FjsSplash (uJsSplsh.pas) เป็นฟอร์มที่ใช้แสดงหน้าต่างไตเติลของเกมส์ Jigsaw Hunter
6. หน้าต่างเกมส์ FjsBoard (uJSBoard.pas) เป็นหน้าต่างที่ใช้แสดงผลการดำเนินการของเกมส์ และบรรจุส่วนของการควบคุมการเล่นเกมส์



รูปที่ 2 แสดงความสัมพันธ์ของแต่ละฟอร์ม

หลักการดำเนินการ

Jigsaw Hunter โปรแกรมเกมส์ที่ประกอบด้วยส่วนดำเนินการหลัก 3 ส่วนคือ

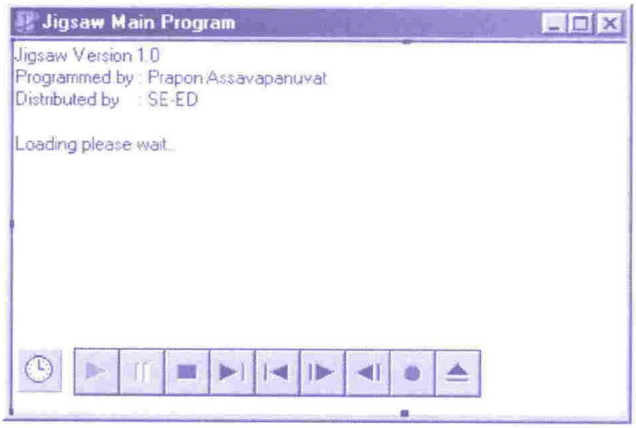
- ส่วนระบบการแสดงผล (Display System) โปรแกรมเกมส์ Jigsaw Hunter มีเทคนิคการแสดงผลโดยการใช้ความสามารถในการแสดงรูปภาพของคอมโปเนนต์ Timage, Tbitmap และ TpaintBox เพื่อแสดงผลบนหน้าจอในลักษณะต่าง ๆ โดยเกมส์นี้ใช้ระบบแสดงผลระดับความละเอียด 1024x768 จุด ที่ 24 บิตสี สำหรับบอร์ดแสดงผล ใช้รูปภาพที่มีความละเอียด 640x480 จุด ในการดำเนินการเล่นเกมส์ โดยแบ่งส่วนของรูปภาพเป็นชิ้นเล็ก ๆ ที่มีขนาดเท่ากันวางสลับไปสลับมาอยู่บนบอร์ดกระดานขนาดใหญ่ เพื่อให้ผู้เล่นสามารถจัดเรียงภาพได้อย่างสะดวก
- ส่วนระบบเพลงประกอบ (Music System) ส่วนนี้ผู้เขียนใช้คอมโปเนนต์ TmediaPlayer ในการควบคุมการเล่นไฟล์ MIDI เป็นเสียงเพลงประกอบภายในเกมส์ ซึ่งสามารถเปลี่ยนแปลงเพลงตามเงื่อนไขที่กำหนด
- ส่วนระบบเสียงเอฟเฟ็คต์ประกอบ (Sound Effects System) สืบเนื่องจากเสียงเอฟเฟ็คต์ต่าง ๆ จะต้องตอบสนองการดำเนินงานต่อเหตุการณ์ที่เกิดขึ้นอย่างรวดเร็วผู้เขียนจึงเลือกใช้วิธีบันทึกเสียงในรูปแบบ Resource File แล้วคอมพายล์ร่วมกันกับโปรแกรม เพื่อสามารถเล่นกลับได้อย่างรวดเร็วด้วยคำสั่งของระบบ WIN32

สำหรับแนวความคิดอื่น ๆ ในการพัฒนาโปรแกรมเกมส์ Jigsaw Hunter นั้นผู้เขียนขออนุญาตกล่าวถึงในแต่ละขั้นตอนตามการพัฒนาที่จะกล่าวถึงต่อไป

ข้อจำกัดและความต้องการของระบบ

โปรแกรม Jigsaw Hunter มีข้อจำกัดและความต้องการของระบบฮาร์ดแวร์ดังต่อไปนี้

- โปรแกรม Jigsaw Hunter สามารถทำงานบนระบบคอมพิวเตอร์ที่มีหน่วยความจำสำหรับระบบแสดงผลไม่ต่ำกว่า 2 MB โดยสามารถปรับระบบแสดงผลให้สามารถแสดงผลด้วยความละเอียดที่ระดับ 1024 x 768 จุด ที่จำนวนสี 16 บิตสีเป็นอย่างน้อย และต้องติดตั้งระบบมัลติมีเดีย เพื่อที่จะสามารถบรรเลงเพลง รวมทั้งเล่นกลับไฟล์เสียงได้
- โปรแกรม Jigsaw Hunter ออกแบบมาให้สามารถเล่นกับหน้าจอความละเอียดขนาด 1024 x 768 เป็นอย่างน้อย โดยโปรแกรมจะติดตั้งระบบตรวจสอบความละเอียดในการแสดงผล ซึ่งถ้าหน้าจอแสดงผลในปัจจุบันน้อยกว่า 1024 จุดจะทำการปรับความละเอียดการแสดงผลให้ตรงตามความต้องการของโปรแกรม



รูปที่ 3 ฟอร์มควบคุมหลัก FjsMain(UjsMain.pas)

- ส่วนไฟล์รูปภาพที่สามารถใช้งานเป็นภาพ Jigsaw จะต้องมีความละเอียดขนาด 640x480 จุด และอยู่ในรูปแบบไฟล์ BMP เท่านั้น โดยไม่จำกัดจำนวนไฟล์ที่ใช้

ฟอร์มควบคุมหลัก FjsMain

ฟอร์มควบคุมหลัก FjsMain (UjsMain.pas) ดังแสดงไว้ในรูปที่ 3 โดยมีส่วนของโปรแกรมแสดงในลิสติงที่ 1 (ติดต่อผ่านคอลัมน์ซอฟต์เมล์) เป็นฟอร์มที่ใช้ควบคุมการทำงานหลักของเกมส์ Jigsaw Hunter ซึ่งเป็นฟอร์มที่ใช้รวบรวมแบบข้อมูล ค่าคงที่ โปรแกรมย่อยต่าง ๆ รวมทั้งส่วนควบคุมระบบเสียงประกอบในเกมส์ สำหรับฟอร์ม FjsMain มีส่วนของโปรแกรมและกิจกรรมที่สำคัญดังต่อไปนี้

ค่าคงที่ (Const)

ยูนิต uJSMain.pas มีรายการค่าคงที่สำหรับใช้ในระบบทั้งหมด (ทุกฟอร์ม) ดังนี้

```
MAX_SCORE          = 12;          // ระบุจำนวนรายชื่อสูงสุดที่สามารถ
                                  บันทึกในรายการผู้เล่นคะแนนสูงสุด
                                  (High Score)
MAX_JIGSAW_PIECE   = 100;         // จำนวนชิ้นภาพ Jigsaw
                                  สูงสุดที่สามารถใช้ได้ในระบบ
JIGSAW_BOARD_LEFT  : integer = 125; // ระบุตำแหน่งแสดงผลของ
                                  บอร์ดต่อภาพ Jigsaw
JIGSAW_BOARD_TOP   : integer = 125;
JIGSAW_BOARD_WIDTH : integer = 640; // ระบุขนาดความกว้างของ
                                  บอร์ดต่อภาพ Jigsaw
JIGSAW_BOARD_HEIGHT : integer = 480; // ระบุขนาดความสูงของ
                                  บอร์ดต่อภาพ Jigsaw

SND_PLAY           = $0001;       // แฟล็กซ์ระบุการเล่นไฟล์เสียง
                                  แบบ Async Mode
SND_WAIT           = $0002;       // แฟล็กซ์ระบุการเล่นไฟล์เสียง
                                  แบบ Sync Mode คือรอให้จบไฟล์
                                  ก่อนดำเนินการต่อ

MUSIC_INTRO        = $0001;       // แฟล็กซ์ระบุการเล่นไฟล์เพลงส่วน
                                  หน้าจอ Splash Screen (Music Intro)
```

```
  MUSIC_LIST          = $0002;                  // แฟล็กซ์ระบุการเล่นไฟล์เพลงจากรายการที่กำหนด
  MUSIC_NAME          = $0003;                  // แฟล็กซ์ระบุการเล่นไฟล์เพลงแบบกำหนดเอง
  MUSIC_HURRY         = $0004;                  // แฟล็กซ์ระบุการเล่นไฟล์เพลงสำหรับส่วนการทำงานแบบเร่งรีบ
  MAX_PREVIEW_TIMER : dword = 3000;             // กำหนดค่ามาตรฐานสำหรับแสดงภาพ Preview เท่ากับ 3 วินาที
  TMR_MAX           : dword = 3600;             // กำหนดค่ามาตรฐานสำหรับเวลาที่ใช้ในการเล่นเกมส์แบบ BASIC เท่ากับ 60 นาที
  TMR_KID           : dword = 300;              // กำหนดค่ามาตรฐานสำหรับเวลาที่ใช้ในการเล่นเกมส์แบบ KID เท่ากับ 5 นาที
  TMR_NORMAL        : dword = 180;              // กำหนดค่ามาตรฐานสำหรับเวลาที่ใช้ในการเล่นเกมส์แบบ NORMAL เท่ากับ 3 นาที
  TMR_EXPERT        : dword = 120;              // กำหนดค่ามาตรฐานสำหรับเวลาที่ใช้ในการเล่นเกมส์แบบ EXPERT เท่ากับ 2 นาที
  GME_MAX_LEVEL     : dword = 7;                // กำหนดค่ามาตรฐานสำหรับการเปลี่ยน Level สำหรับเกมส์แบบ BASIC เท่ากับ 7 ภาพ
  GME_KID_LEVEL     : dword = 6;                // กำหนดค่ามาตรฐานสำหรับการเปลี่ยน Level สำหรับเกมส์แบบ KID เท่ากับ 6 ภาพ
  GME_NORMAL_LEVEL  : dword = 5;                // กำหนดค่ามาตรฐานสำหรับการเปลี่ยน Level สำหรับเกมส์แบบ NORMAL เท่ากับ 5 ภาพ
  GME_EXPERT_LEVEL  : dword = 4;                // กำหนดค่ามาตรฐานสำหรับการเปลี่ยน Level สำหรับเกมส์แบบ EXPERT เท่ากับ 4 ภาพ
  JS_PATH : STRING = 'C:\JIGSAW';               // กำหนดไดเรคทอรีที่ใช้บันทึกไฟล์ Configuration
  JS_CONFIG_FILE : STRING = 'JSCONFIG.BIN';     // กำหนดชื่อไฟล์ Configuration ที่ใช้บันทึกค่าเริ่มต้นของเกมส์และรายชื่อผู้เล่น
```

แบบข้อมูล (Type)

ยูนิต UjsMain.pas มีแบบข้อมูลที่สร้างขึ้นเพื่อใช้บันทึกข้อมูลสำหรับการดำเนินงานของเกมส์ที่ใช้ในระบบทั้งหมด (ทุกฟอร์ม) ดังนี้

```
TpictureListR = record
  filename : shortString;  // ชื่อไฟล์รูปภาพ
  picId    : dword;        // หมายเลข ID ของรูปใช้สำหรับสุ่มการจัดเรียงรูปภาพ
end;
TpictureListP = ^TpictureListR;
```

แบบข้อมูลที่ใช้บันทึกรายชื่อภาพที่จะใช้ในการเล่นเกมส์ เพื่อบรรจุลงในรายการลิสต์ (TLIST) ของระบบ

```
TscoreBoardR = record
  name  : string[30];      // ชื่อผู้เล่นเกมส์
  score : dword;           // จำนวนคะแนนที่ทำได้
end;
TscoreBoardS = array[ 0..MAX_SCORE-1 ] of TscoreBoardR;     // อะเรย์บันทึกรายชื่อผู้เล่น
```

แบบข้อมูลสำหรับบันทึกชื่อผู้เล่นที่มีจำนวนคะแนนสูงเป็น 12 อันดับแรก

```
TjsConfigR = record
  preview_tmr   : dword;              // ค่า Preview Timer ที่ใช้ในระบบ
  timer_max     : dword;              // ค่าเวลาที่ใช้สำหรับการเล่นเกมส์แบบ BASIC
  timer_kid     : dword;              // ค่าเวลาที่ใช้สำหรับการเล่นเกมส์แบบ KID
  timer_normal  : dword;              // ค่าเวลาที่ใช้สำหรับการเล่นเกมส์แบบ NORMAL
  timer_expert  : dword;              // ค่าเวลาที่ใช้สำหรับการเล่นเกมส์แบบ EXPERT
  gme_max_lev   : dword;              // จำนวนภาพที่ใช้เพื่อเปลี่ยน Level ถัดไปสำหรับ BASIC GAME
  gme_kid_lev   : dword;              // จำนวนภาพที่ใช้เพื่อเปลี่ยน Level ถัดไปสำหรับ KID GAME
  gme_norm_lev  : dword;              // จำนวนภาพที่ใช้เพื่อเปลี่ยน Level ถัดไปสำหรับ NORMAL GAME
  gme_exp_lev   : dword;              // จำนวนภาพที่ใช้เพื่อเปลี่ยน Level ถัดไปสำหรับ EXPERT GAME
  topScoreBoard : TscoreBoardS;       // รายชื่อผู้เล่นที่มีคะแนนสูงสุด 12 คน
end;
```

แบบข้อมูลสำหรับบันทึกค่าเริ่มต้น Configuration ที่ใช้ในระบบ

```
TimgPieceR = record
  id          : integer;      // หมายเลข ID ประจำชิ้นภาพ
  x1,y1,x2,y2 : integer; // ตำแหน่งชิ้นส่วนของภาพจริง โดยระบุตำแหน่งของมุมซ้าย มุมบน มุมขวาและมุมล่างของชิ้นภาพ
end;
TimgPieceS = array[ 0..MAX_JIGSAW_PIECE-1 ] of TimgPieceR; // อะเรย์ที่บันทึกชิ้นภาพทั้งหมด
```

แบบข้อมูลสำหรับบันทึกตำแหน่งชิ้นส่วนของภาพที่นำไปสร้างเป็นชิ้นภาพ เพื่อใช้ระบุตำแหน่งจริงของชิ้นภาพแต่ละชิ้น

```
  TjigsawPieceR = record
  id          : integer;      // ID ประจำภาพเป็นหมายเลขเดียวกันกับค่าที่บันทึกอยู่ในพร็อปเพอร์ตี้ picture.tag เพื่อใช้อ้างถึงกัน
  picture     : TImage;  // ภาพชิ้นส่วนตามขนาด x1,y1,x2,y2 บันทึกในรูปแบบ TImage
  x1,y1,x2,y2 : integer;  // ตำแหน่งชิ้นส่วนของภาพที่คัดลอกจากภาพจริง
  gridValue   : integer;  // ระบุตำแหน่งช่องของชิ้นภาพที่นำไปวาง (ปกติมีค่าเท่ากับ -1 คือไม่ตรงกับช่องใด)
end;
TjigsawPieceS = array[ 0..MAX_JIGSAW_PIECE-1 ] of TjigsawPieceR; //อะเรย์ของชิ้นภาพที่ใช้ในเกมส์
```

แบบข้อมูลสำหรับบันทึกชิ้นภาพ Jigsaw แต่ละชิ้นที่ใช้ในเกมส์เพื่อต่อเป็นภาพจริง

ตัวแปร

ยูนิต uJSMain.pas มีตัวแปรที่ใช้บันทึกข้อมูลสำหรับการดำเนินงานของเกมส์ ดังนี้

```
tmrCnt : integer;        // ตัวแปรสำหรับจัดลำดับการทำงานของ Timer เพื่อเริ่ม
                         ดำเนินการเกมส์
musicType : word;        // ตัวแปรสำหรับระบุชนิดเพลงที่กำลังดำเนินการ
musicGameCnt : word;     // ตัวแปรระบุลำดับเพลงที่บรรเลงในส่วนการเล่นเกมส์
jsConfig : TjsConfigR;   // ตัวแปรบันทึกค่า Configuration ของระบบที่ผู้เล่น
                         กำหนดตามความเหมาะสม
pictureList : TList;     // รายชื่อไฟล์ภาพที่ใช้เล่นเกมส์ โดยอ่านจากไดเรคทอรี
                         ..\picture ของไดเรคทอรีปัจจุบัน
```

Resource File

ยูนิต UjsMain.pas มีส่วนของโปรแกรมเพิ่มเติมเพื่อระบุการเรียก Resource File เพื่อคอมพายล์ (Compile) ร่วมกันดังนี้

```
{$R C:\JIGSAW\WAVE\JSSOUND.RES}  // ระบุการเรียกใช้ Resource File
                                 ด้วยคอมพลาย์เลอร์ไดเร็คทีฟ $R
```

- function MakeDword(A, B: Word): dword; เป็นฟังก์ชันสำหรับสร้างข้อมูลชนิด Dword ด้วยข้อมูลชนิด Word จำนวน 2 ชุดสำหรับกำหนดค่าระดับเสียงให้แก่คำสั่ง MidioutSetVolume และ WaveOutSetVolume เพื่อปรับระดับเสียง
- function GetRandomValByTicker(mxValue : dword) : dword; ฟังก์ชันสำหรับสุ่มตัวเลขโดยคำนวณจากค่า TickCount ของระบบวินโดวส์ โดยค่าที่ได้จะอยู่ระหว่าง 0 ถึง mxValue-1
- procedure Delay(msec : dword); โปรแกรมย่อยสำหรับดำเนินการหน่วงเวลาเท่ากับ msec มิลลิวินาที
- procedure GetDisplayRes(var width,height : integer); โปรแกรมย่อยสำหรับอ่านค่าความละเอียดของระบบแสดงผลของหน้าจอเดสค์ท้อป (Desktop) ในปัจจุบัน
- procedure SetDisplay(bitColor : integer; cWidth, cHeight : integer; freq : integer); โปรแกรมย่อยสำหรับเปลี่ยนแปลงความละเอียดในการแสดงผลของระบบแสดงผลปัจจุบันให้เป็นความละเอียดในการแสดงผลใหม่ โดยระบุจำนวนบิตสีที่ใช้ด้วย bitColor, ความกว้าง cWidth, ความสูง cHeight และด้วยความถี่ freq โดยการเรียกใช้คำสั่ง ChangeDisplaySettings ของระบบ WIN32
- procedure TFjsMain.DoInitialized; กิจกรรมสำหรับกำหนดค่าเบื้องต้นให้กับโปรแกรมเกมส์ Jigsaw Hunter และเริ่มต้นรันโปรแกรมโดย

```
SetupMusicDev( $7FFF,$7FFF );     // กำหนดระดับเสียงเพลง
SetupSoundDev( $FFFF,$FFFF );     // กำหนดระดับเสียงเอฟเฟ็คต์
                                  ที่ใช้ในเกมส์
pictureList:=TList.Create;        // สร้าง PictureList เพื่อใช้บรรจุรายชื่อไฟล์
                                  รูปภาพ
pictureList.Clear; pictureList.Capacity:=1000;
ret:=LoadPictureFileList('.\PICTURE\*.BMP'); // อ่านชื่อไฟล์รูปภาพจาก
                                  ไดเรคทอรีที่กำหนด
DoPlayMusicIntro;                 // เริ่มบรรเลงเพลง Intro
```

- procedure TFjsMain.DoPlaySplashScreen; กิจกรรมสำหรับแสดงหน้าต่าง Splash Screen เพื่อแสดงโลโก้ของโปรแกรมเกมส์ Jigsaw Hunter โดยเรียกกิจกรรม FjsSplash.DisplaySplashScreen เพื่อแสดงภาพเอฟเฟ็คต์
- procedure TFjsMain.DoPlayMusicIntro; กิจกรรมสำหรับดำเนินการบรรเลงเพลง Intro และกล่าวคำแนะนำเกมส์ โดยเรียกใช้กิจกรรม PlayMusic และ MakeSound ตามลำดับดังนี้

```
PlayMusic( JS_MUSIC_INTRO, MUSIC_INTRO );  // บรรเลงเพลงชื่อไฟล์
                                            JS_MUSIC_INTRO
with musicPlayer do                         // วนลูปรอกระทั่งเพลงเริ่มบรรเลง
  repeat schedule; until ( mode = mpPlaying );
MakeSound( 'GAME', SND_PLAY ); /            // กล่าวคำแนะนำด้วยการเรียกใช้
                                            ข้อมูลเสียงที่บันทึกอยู่ใน
                                            Resource ชื่อ GAME
```

- function TFjsMain.IsMusicPlaying : boolean; กิจกรรมชนิดฟังก์ชันสำหรับตรวจสอบการทำงานของระบบเสียงดนตรี โดยให้ค่าเท่ากับ TRUE ถ้ากำลังใช้งาน
- procedure TFjsMain.WaitMusicStop; กิจกรรมสำหรับวนลูปรอจนจบการบรรเลงเพลงในปัจจุบัน
- procedure TFjsMain.PlayMusic(fname: shortString; mType : word); กิจกรรมสำหรับกำหนดการเริ่มบรรเลงด้วยชื่อไฟล์ fname และชนิด mType ดังนี้

```
with musicPlayer do
begin
  musicType:=mType;    // กำหนดชนิดไฟล์ที่บรรเลง
  if ( mode in [mpPlaying, mpSeeking, mpOpen] ) then
  begin stop; close; end; // ถ้ามีการบรรเลงอยู่แล้วให้หยุด และปิดมีเดีย
  fileName:=fname; notify:=true; open; Play; // กำหนดชื่อไฟล์และเปิดไฟล์เพื่อ
                                             บรรเลงเพลง
end;
```

- procedure TFjsMain.MakeSound(sndNameP : pchar; sndFlg : word); กิจกรรมสำหรับเล่นกลับไฟล์เสียงหรือเอฟเฟ็คต์ที่บันทึกอยู่ใน Resource (.RES) โดยระบุชื่ออ้างถึง (Alias) ไว้ในพอยน์เตอร์ sndNameP และชนิดการเล่นใน sndFlg โดยถ้า sndFlg เป็น SND__PLAY หมายถึงเล่นแบบ Async Mode และ SND__WAIT หมายถึงเล่นแบบ Sync Mode คือรอจนจบไฟล์แล้วจึงดำเนินการคำสั่งต่อไป
- procedure TFjsMain.StopSound; กิจกรรมสำหรับบังคับให้หยุดเล่นกลับไฟล์เสียงทันที โดยผ่านค่า Nil ให้กับชื่อที่อ้างถึงใน Resource File ดังนั้น

```
PlaySound( nil, hInstance, snd_ASync or snd_Resource );
```

- procedure TFjsMain.DoPlayMusicGame; กิจกรรมสำหรับบรรเลงเพลงที่ใช้ในการเล่นเพลงจากลิสต์รายการเพลงที่เตรียมไว้
- procedure TFjsMain.AddPictureToList(p: TpictureListP); กิจกรรมสำหรับเพิ่มชื่อรูปภาพที่ต้องการใช้ในการเล่นเกมส์เข้าสู่ลิสต์รายการภาพ

- procedure TFjsMain.ReleasePictureList: กิจกรรมสำหรับดำเนินการยกเลิกหน่วยความจำที่บรรจุรายชื่อรูปภาพที่ใช้ในโปรแกรมทั้งหมด
- procedure TFjsMain.InitVar: กิจกรรมสำหรับกำหนดค่าเบื้องต้นของโปรแกรม Jigsaw Hunter ในส่วนการอ่านข้อมูล Configuration และกำหนดความละเอียดของหน้าจอแสดงผลให้ได้ขนาดตามต้องการ

```
// ถ้าไม่มีไดเรคทอรี JS_PATH นี้ในไดรฟ์ C:\ ให้ทำการสร้าง
if ( not directoryExists( JS_PATH ) ) then ForceDirectories( JS_PATH );
LoadConfig;                    // อ่าน Configuration จากไฟล์
GetDisplayRes( wid, hi );// อ่านค่าความละเอียดในการแสดงผลเดิม
if ( wid<1024 ) then  SetDisplay( 16, 1024, 768, 60 ); // ปรับเป็นความละเอียด
                                                       ในการแสดงผลที่ต้องการ
```

- procedure TFjsMain.SetupMusicDev(Llevel, Rlevel : word): กิจกรรมสำหรับกำหนดระดับเสียงของการบรรเลงเพลง MIDI ไฟล์ด้วยคำสั่ง midiOutSetVolume ดังนี้

```
for c:= 0 to midiOutGetNumDevs -1 do// กำหนดให้อ่านค่า midi device
                                    จากระบบทั้งหมด
begin
                                    // อ่านค่ามาตรฐานที่กำหนดให้ midi device
  MidiOutGetDevCaps( c, @midiOutCaps, sizeof( TMidiOutCapS ) );
  // ตรวจสอบ ถ้าสามารถปรับระดับเสียงได้ให้ทำการกำหนดระดับเสียงใหม่
  if ( MidiOutCaps.dwSupport and MIDICAPS_LRVOLUME <> 0 ) then
    midiOutSetVolume( c, MAKEDWORD( Llevel, Rlevel ) ); // กำหนดระดับเสียง
                                                        เพลงใหม่
end;
```

- procedure TFjsMain.SetupSoundDev(Llevel, Rlevel : word): กิจกรรมสำหรับกำหนดระดับเสียง Wave ด้วยคำสั่ง

```
for c:= 0 to waveOutGetNumDevs - 1 do   // วงรอบเพื่อตรวจสอบ Wave Device
                                        ที่ติดตั้งอยู่ในระบบ
begin
      // อ่านค่ามาตรฐานที่กำหนดให้ Wave Device ที่เลือก
  WaveOutGetDevCaps( c, @wavOutCaps, sizeof( TWaveOutCaps ) );
      // ถ้า Wave Device มีคุณสมบัติการปรับเสียงให้ทำการปรับเสียงด้วยระดับ LLevel
         และ RLevel
  if ( wavOutCaps.dwSupport and WAVECAPS_LRVOLUME <> 0 ) then
    waveOutSetVolume( c, MAKEDWORD( Llevel, Rlevel ) );
end;
```

- procedure TFjsMain.LoadConfig: กิจกรรมสำหรับอ่านข้อมูล Configuration จากไฟล์ JS__CONFIG__FILE
- procedure TFjsMain.SaveConfig: กิจกรรมสำหรับบันทึกข้อมูล Configuration ลงไฟล์ JS__CONFIG__FILE
- function TFjsMain.LoadPictureFileList(path : shortString) : integer: กิจกรรมสำหรับค้นหาไฟล์รูปภาพจากไดเรคทอรี path ที่กำหนดดังนี้

```
// เริ่มค้นหาไฟล์ใน Path ที่กำหนด
ret:=FindFirst( path, faAnyFile, SearchRec ); cnt:=0; result:=cnt;
  // ถ้าไม่พบให้ยกเลิก
if ( ret<>0 ) then begin FindClose( SearchRec ); exit; end;
 fname:=uppercase( searchRec.name ); // กำหนดให้ชื่อไฟล์เป็นตัวใหญ่ทั้งหมด
if ( pos( '.BMP', fname )>0 ) then // ตรวจสอบว่ามี File Type เป็น BMP หรือไม่
begin
        // บันทึกชื่อไฟล์ลงลิสต์รายการไฟล์รูปภาพ
  p:=allocMem( Sizeof( TpictureListR ) ); p^.filename:=fname;
  AddPictureToList( p ); inc( cnt );
end;
while ( ret=0 ) do // ดำเนินการวงรอบเพื่อค้นหาไฟล์อื่น
begin
  ret:=FindNext( SearchRec ); // อ่านข้อมูลของไฟล์ต่อไป
  if ( ret=0 ) then // ถ้าพบไฟล์ใหม่
  begin
    fname:=uppercase( searchRec.name );
    if ( pos( '.BMP', fname )>0 ) then  // ตรวจสอบว่ามี File Type เป็น BMP หรือไม่
    begin
              // ถ้าใช่ให้บันทึกลงลิสต์รายการไฟล์รูปภาพ
      p:=allocMem( Sizeof( TpictureListR ) ); p^.filename:=fname;
      AddPictureToList( p ); inc( cnt );
    end;
  end;
end; // กลับไปดำเนินการต่อไป ถ้าพบไฟล์ในไดเรคทอรีนี้
FindClose( SearchRec );
```

- procedure TFjsMain.DoQuickSortFile: กิจกรรมสำหรับเรียงลำดับข้อมูล (sort) ด้วยอัลกอริทึม Quick Sort ของเดลไฟ
- procedure TFjsMain.DoShuffleFileList: กิจกรรมสำหรับสุ่มตัวเลขเพื่อสลับตำแหน่งรูปภาพทุกครั้งที่เรียกใช้กิจกรรมนี้โดย

```
maxPicCnt:=pictureList.count; // อ่านจำนวนรูปภาพในลิสต์รายการทั้งหมด
delayTime:=5000 div maxPicCnt; // คำนวณหาค่า delayTime เพื่อใช้กำหนด
                                  การหน่วงเวลาไม่เกิน 5 วินาทีสำหรับข้อมูลทั้งหมด
randomize; // เปลี่ยนชุดตัวเลขสุ่ม
for c:= 0 to maxPicCnt-1 do // ทำการสุ่มลำดับใหม่ให้กับรูปภาพทุกรูปในลิสต์
begin
  p:=pictureList.items[c];  p^.picId:=GetTickCount mod (maxPicCnt+1); //
เปลี่ยนลำดับภาพใหม่ให้กับรูป
  delay( random( delayTime ) ); // หน่วงเวลาเท่ากับจำนวนสุ่มที่ไม่เกิน delayTime
end;
DoQuickSortFile; // เรียงลำดับข้อมูลใหม่
```

- function TFjsMain.GetPictureFromList(id: integer): shortString: กิจกรรมสำหรับอ่านชื่อไฟล์จากลำดับ id ในลิสต์รายการไฟล์รูปภาพ
- function TFjsMain.GetScoreLevel(newScore : dword) : integer: กิจกรรมสำหรับหาตำแหน่งอะเรย์ที่จะบันทึกคะแนนจากลิสต์รายการคะแนนเดิม โดยเปรียบเทียบกับคะแนนที่อยู่ในลิสต์เดิม
- procedure TFjsMain.AddNewScore(newName: shortString; newScore: dword): กิจกรรมสำหรับเพิ่มรายชื่อผู้เล่นรายใหม่ที่มีคะแนนสูงกว่าที่บันทึกไว้ในลิสต์รายชื่อผู้เล่น โดยระบบจะทำการตรวจสอบว่าผู้เล่นใหม่มีคะแนนมากกว่าที่บันทึกหรือไม่ ถ้ามากกว่าจะทำการบันทึก ณ ตำแหน่งที่มากกว่า

```
// หาตำแหน่งที่จะเพิ่มรายการผู้เล่นใหม่ โดยเปรียบเทียบกับคะแนนในลิสต์รายการเดิม
sNum:=GetScoreLevel( newScore ); if ( sNum<0 ) then exit; // ถ้าคะแนนน้อยกว่าให้ยกเลิกการเพิ่มรายการ
if ( sNum>MAX_SCORE-1 ) then
begin // เลื่อนข้อมูลตำแหน่งที่จะแทรกลงไป 1 เรคอร์ด
  move( jsConfig.topScoreBoard[ sNum ], jsConfig.topScoreBoard[ sNum+1 ],
  ( (MAX_SCORE-1) - snum ) * sizeof( TscoreBoardS ) );
end;
with jsConfig.topScoreBoard[ sNum ] do // แทรกข้อมูลในเรคอร์ดตำแหน่งที่ตรวจพบ
begin  name:=newName; score:=newScore; end;
```

- procedure TFjsMain.MusicPlayerNotify(Sender: TObject): กิจกรรมสำหรับตรวจสอบการบรรเลงเพลง โดยถ้าบรรเลงจบเพลงแล้วให้ดำเนินการตามเงื่อนไขที่กำหนด เช่น บรรเลงเพลงซ้ำหรือบรรเลงเพลงต่อไป เป็นต้น

การสร้าง Resource File

เดลไฟได้จัดเตรียมโปรแกรมชนิด Command line ชื่อ BRCC32.EXE ที่ติดตั้งอยู่ในไดเรคทอรีเดียวกันกับโปรแกรมเดลไฟ เป็นโปรแกรม Resource File Compiler ใช้สำหรับสร้าง Resource File (.RES) วิธีใช้งานทำได้โดย

1. สร้างไฟล์รายการ Resource List ด้วยเท็กซ์เอดิเตอร์ทั่วไป เช่น NotePad

2. ไฟล์ Resource List จะประกอบด้วยฟิลด์ข้อมูลดังต่อไปนี้

```
<Alias Name> <Resource Type> <File Name>
        โดย :
        Alias Name      หมายถึงชื่อที่ใช้อ้างถึง Resource <File Name>
        Resource Type หมายถึงชนิดของ Resource ที่จะทำการบันทึก
        File Name       คือชื่อไฟล์ข้อมูลมีรูปแบบที่กำหนดตาม Resource Type
                        เช่น
        BLEEP   WAVE   BLEEP.WAV
```

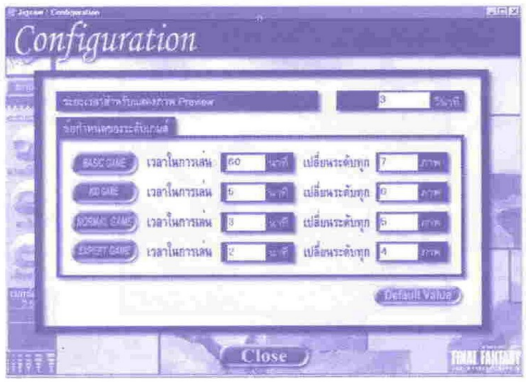
3. บันทึกรายการไฟล์ที่ต้องการสร้างเป็น Resource File ทั้งหมดลงแฟ้มข้อมูล ตัวอย่างของไฟล์รายการ Resource ที่ใช้แสดงอยู่ในลิสติงที่ 2 (ติดต่อผ่านคอลัมน์ซอฟต์เมล์)

4. ทำการคอมพายล์แฟ้มข้อมูลที่สร้างด้วย Resource Compiler ดังนี้

```
BRCC32 <fileName>
```

5. จะได้ผลลัพธ์เป็น <fileName>.RES เพื่อนำไปลิงก์ใช้งานร่วมกันกับโปรแกรมเดลไฟด้วยคอมพายเลอร์ไดเรคทีฟ $R ดังนี้

```
{$R C:\JIGSAW\WAVE\<fileName>.RES}
```



รูปที่ 4 ฟอร์มหน้าต่างกำหนดรายละเอียดเบื้องต้น FjsConfig (UjsConfig.pas)

ฟอร์มหน้าต่างกำหนดรายละเอียดเบื้องต้น FjsConfg

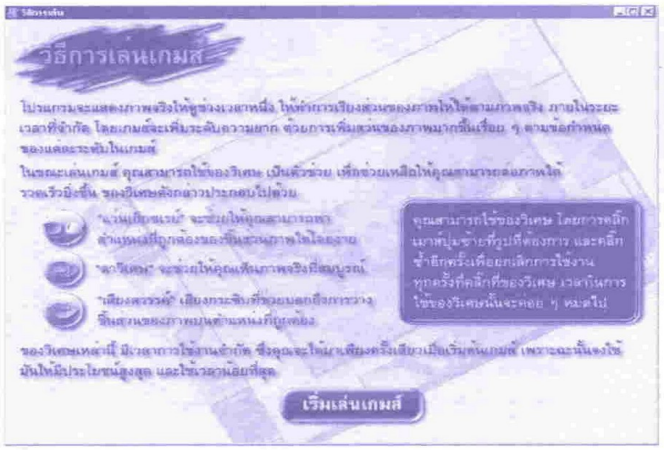
สำหรับฟอร์มหน้าต่างกำหนดรายละเอียดเบื้องต้น FjsConfg (uJSConfg.pas) ดังรูปที่ 4 ส่วนของโปรแกรมจะอยู่ในลิสติงที่ 3 (ติดต่อผ่านคอลัมน์ซอฟต์เมล์) เป็นฟอร์มที่ใช้ควบคุมการกำหนดรายละเอียดเบื้องต้น เพื่อปรับความเหมาะสมในการเล่นเกมส์สำหรับผู้เล่นแต่ละระดับขั้น ฟอร์ม FjsConfg มีส่วนของโปรแกรมและกิจกรรมที่สำคัญดังต่อไปนี้

- procedure TFjsConfg.LoadConfig: เป็นกิจกรรมสำหรับอ่านข้อมูลจากตัวแปร FjsMain.jsConfig เพื่อแสดงผลบนฟอร์ม เตรียมให้ผู้เล่นเกมส์ปรับค่าต่าง ๆ ดังนี้

```
with FjsMain.jsConfig do // แสดงข้อมูลที่บันทึกอยู่ใน FjsMain.jsconfig
begin
  edit1.text:=IntToStr( preview_tmr div 1000 );
  edit2.text:=IntToStr( timer_max div 60 );
  edit3.text:=IntToStr( gme_max_lev );
  edit4.text:=IntToStr( timer_kid div 60 );
  edit5.text:=IntToStr( gme_kid_lev );
  edit6.text:=IntToStr( timer_normal div 60 );
  edit7.text:=IntToStr( gme_norm_lev );
  edit8.text:=IntToStr( timer_expert div 60 );
  edit9.text:=IntToStr( gme_exp_lev );
end;
end;
```

- procedure TFjsConfg.SaveConfig: กิจกรรมสำหรับอ่านข้อมูลจากหน้าต่างกำหนดรายละเอียดเบื้องต้น FjsConfg เข้าสู่ตัวแปร FjsMain.jsConfig เพื่อใช้ในการควบคุมโปรแกรม Jigsaw Hunter และเก็บไว้เป็นค่าเริ่มต้นสำหรับใช้งานในครั้งต่อไปดังนี้

```
with FjsMain.jsConfig do
begin // อ่านข้อมูลจาก Edit Box แล้วเปลี่ยนค่าเป็นจำนวนเต็มเพื่อใช้งานในเกมส์
  preview_tmr:=StrToIntDef( edit1.text, (MAX_PREVIEW_TIMER div 1000) ) * 1000;
  timer_max:=StrToIntDef( edit2.text, (TMR_MAX div 60) ) * 60;
  gme_max_lev:=StrToIntDef( edit3.text, GME_MAX_LEVEL );
  timer_kid:=StrToIntDef( edit4.text, (TMR_KID div 60) ) * 60;
  gme_kid_lev:=StrToIntDef( edit5.text, GME_KID_LEVEL );
  timer_normal:=StrToIntDef( edit6.text, (TMR_NORMAL div 60) ) * 60;
  gme_norm_lev:=StrToIntDef( edit7.text, GME_NORMAL_LEVEL );
  timer_expert:=StrToIntDef( edit8.text, (TMR_EXPERT div 60) ) * 60;
  gme_exp_lev:=StrToIntDef( edit9.text, GME_EXPERT_LEVEL );
end;
```



รูปที่ 5 ฟอร์มรายละเอียดเบื้องต้น FjsConfig (UjsConfig.pas) ฟอร์มหน้าต่างช่วยเหลือ FjsHelp

ฟอร์มหน้าต่างช่วยเหลือ FjsHelp

ฟอร์มหน้าต่างช่วยเหลือ FjsHelp (uJSHelp.pas) ดังรูปที่ 5 เป็นฟอร์มที่ใช้แสดงคำอธิบายวิธีการเล่นเกมส์ Jigsaw Hunter เพื่อช่วยให้ผู้เล่นเกมส์หน้าใหม่สามารถทำความเข้าใจกฎและกติกาการเล่นก่อนเริ่มเล่นเกมส์จริงได้ สำหรับฟอร์มนี้ผู้เขียนขอไม่แสดงส่วนของโปรแกรม เนื่องจากไม่มีส่วนของโปรแกรมใด ๆ ที่สำคัญ

จบเกมส์แรก

สำหรับการเริ่มต้นพัฒนาโปรแกรมเกมส์ผู้อ่านจะพบว่าสามารถเริ่มต้นได้ไม่ยากจนเกินไป (ถึงแม้มีอุปสรรคบ้าง) ฉบับนี้ผู้เขียนต้องขอจบตอนของการสร้างฟอร์มทั้งสามเพียงเท่านี้ ฉบับหน้าผู้เขียนจะกล่าวถึงแนวความคิดในการดำเนินงานที่เกิดขึ้นภายในเกมส์ Jigsaw Hunter เพื่อสร้างฟอร์มดำเนินการเกี่ยวกับเกมส์โดยตรงและส่วนของฟอร์มต่าง ๆ ที่เหลือ

สำหรับผู้อ่านที่มีข้อสงสัยเกี่ยวกับส่วนใด ๆ ของการพัฒนาโปรแกรมนี้ สามารถส่งอีเมล์สอบถามได้ที่ prapona@hotmail.com

ผู้อ่านท่านใดสนใจ Source Code ของโปรแกรมนี้สามารถติดต่อผ่านคอลัมน์ซอฟต์เมล์ได้

Micro Short Tips

เทคนิคในการใช้ฟิลเตอร์ของ Photoshop

ฟิลเตอร์ถือเป็นหนึ่งในคุณสมบัติและความสามารถที่โดดเด่นของ Photoshop อย่างไรก็ตามการเรียนรู้การใช้ฟิลเตอร์เหล่านี้จำเป็นต้องอาศัยวิธีลองผิดลองถูกเพื่อให้ได้ผลลัพธ์ใหม่ที่ต้องการทุก ๆ ครั้งที่คุณทดลองใช้ฟิลเตอร์ใด ๆ อย่าลืมว่าหลังจากใช้แล้วคุณยังสามารถปรับแต่งผลของเอฟเฟ็กต์ที่สร้างขึ้นได้ด้วยการเลือกคำสั่ง Filter > Fade Filter จากนั้นปรับแต่งค่าต่าง ๆ ของฟิลเตอร์ได้ตามที่คุณต้องการ เช่น ปรับให้ภาพเบลอมากขึ้นหรือน้อยลง ลดหรือเพิ่มระดับความทึบแสง ฯลฯ

คำแนะนำในการตั้งรหัสผ่านเพื่อความปลอดภัย

วิธีที่ดีในการสร้างความปลอดภัยให้กับเครื่องคอมพิวเตอร์ต้องประกอบด้วยปัจจัยหลายอย่าง หนึ่งในนั้นที่หลายคนละเลยคือการตั้งพาสเวิร์ดหรือรหัสผ่านสำหรับการล็อกอินเข้าเครือข่าย การตั้งรหัสผ่านต้องใช้ชื่อที่ยากแก่การเดาหรือวิเคราะห์ รหัสผ่านนั้นควรจะประกอบตัวตัวอักษรไม่ต่ำกว่า 8 ตัว ผสมผสานระหว่างตัวอักษรทั้งพิมพ์ใหญ่-พิมพ์เล็ก ตัวเลข สัญลักษณ์ รหัสผ่านไม่ควรซ้ำหรือใกล้เคียงกับรหัสที่เคยใช้ก่อนหน้านี้ และไม่ควรประกอบด้วยชื่อ-นามสกุลของตัวเอง สัตว์เลี้ยง เพื่อนสนิท ญาติพี่น้อง บุคคลในครอบครัว ดาราหรือนักร้องที่ชื่นชอบซึ่งใคร ๆ ต่างทราบกันดีและไม่ควรใช้คำทั่ว ๆ ไป

ของฟรีที่คุณหาได้ในเว็บ

ทุกคนชอบของฟรี คุณทราบหรือไม่ว่ามีเว็บไซต์ชื่อ Free Stuff Center (http://www.freestuffcenter.com) สถานที่ซึ่งคุณจะพบกับของฟรีต่าง ๆ ไม่ว่าจะเป็นคูปอง นิตยสาร ตัวอย่างสินค้า รวมถึงซอฟต์แวร์คอมพิวเตอร์

อย่างไรก็ตามของฟรีไม่มีในโลก สิ่งของเหล่านี้ต้องแลกเปลี่ยนกับชื่อ ที่อยู่ อีเมล์ ฯลฯ ซึ่งจะเก็บไว้ในฐานข้อมูล ประเด็นนี้อาจไม่ใช่เรื่องเลวร้าย แต่อย่าลืมว่าคุณอาจจะต้องพบกับจดหมายโฆษณาขายของจำนวนมากที่พร้อมจะส่งถึงมือคุณในเวลาอันรวดเร็ว

แต่ปัญหาที่ว่านี้สามารถป้องกันได้โดยใช้ตัวกรองในโปรแกรมรับส่งอีเมล์ของคุณ

เทคนิคทำนัดจากรายการติดต่อใน Outlook

คุณสามารถสร้างการนัดหมายจากรายการติดต่อ (Contacts) อย่างรวดเร็ว ด้วยการคลิกแล้วลากรายชื่อบุคคลที่คุณต้องการติดต่อไปยังโฟลเดอร์ Calendar ทางซ้ายมือ ไดอะล็อกบ็อกซ์ Meeting จะปรากฏขึ้น เพื่อให้คุณป้อนข้อมูลการนัดหมายต่าง ๆ ได้ตามต้องการ